Apprentissage supervisé

Introduction à l'apprentissage automatique – GIF-4101 / GIF-7005

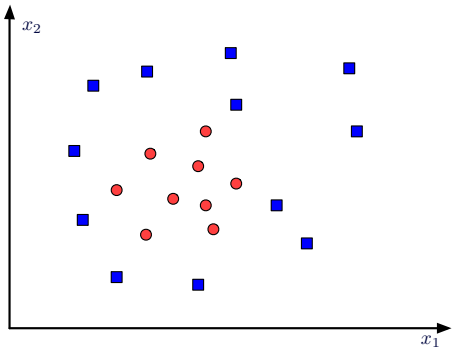
Professeur : Christian Gagné

Semaine 1

Apprendre à partir d'exemples

- Supposons une classe correspondant au concept de *voiture familiale*
- Problème à deux classes
  - Positif (cercles rouges) : est une voiture familiale
  - Négatif (carrés bleus) : n'est pas une voiture familiale
- Représentation des exemples sur deux dimensions
  - x_1 : prix de la voiture
  - x_2 : puissance du moteur

Apprendre à partir d'exemples



- Exemples :

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

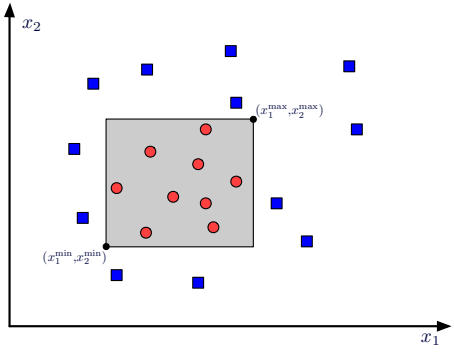
- Étiquettes de classe :

$$r = \begin{cases} 1 & \text{si } \mathbf{x} \text{ est positif} \\ 0 & \text{si } \mathbf{x} \text{ est négatif} \end{cases}$$

- Jeu de N exemples :

$$\mathcal{X} = \{\mathbf{x}^t, r^t\}_{t=1}^{N}$$

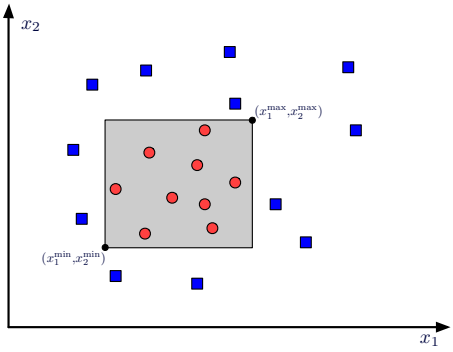
Hypothèse de classement



- Hypothèse possible :

$$(x_1^{\min} \leq x_1 \leq x_1^{\max}) \text{ et } (x_2^{\min} \leq x_2 \leq x_2^{\max})$$

Classe d'hypothèses



- Hypothèse particulière : $\mathrm{h} \in \mathcal{H}$

$$\mathrm{h}(\mathbf{x}) = \begin{cases} 1 & \text{si h classe } \mathbf{x} \\ & \text{positif} \\ 0 & \text{si h classe } \mathbf{x} \\ & \text{négatif} \end{cases}$$

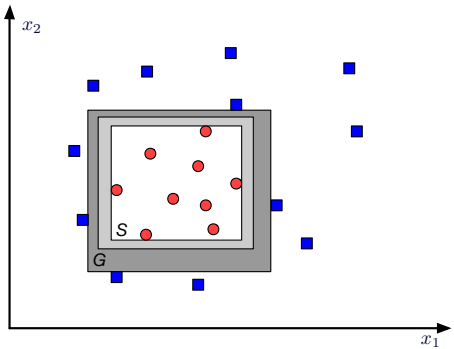
- Erreur empirique :

$$E(\mathrm{h}|\mathcal{X}) = \frac{1}{N}\sum_{t=1}^{N} \mathcal{L}(\mathrm{h}(\mathbf{x}^t), r^t)$$

- Fonction de perte 0-1 :

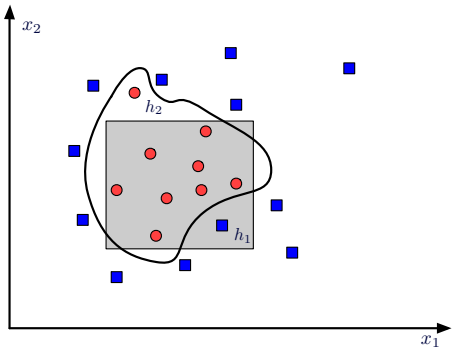
$$\mathcal{L}(a,b) = \begin{cases} 1 & \text{si } a \neq b \\ 0 & \text{si } a = b \end{cases}$$

Hypothèses générales et spécifiques



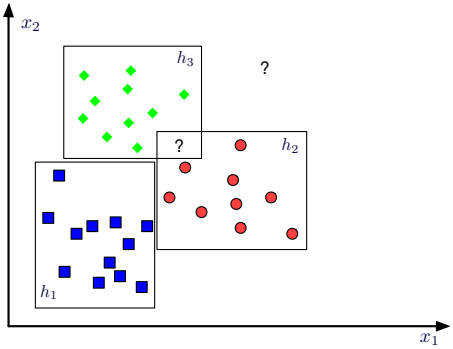
- G : hypothèse la plus générale
- S : hypothèse la plus spécifique
- Hypothèses dans $\mathcal{H}$ entre S et G font parties de l'*espace des versions*

Complexité des modèles et bruit



- Bruit dans les données
  - Manque de précision
  - Erreurs d'étiquetage
  - Mesures latentes
- À performances égales, préférer le modèle le plus simple
  - Complexité : plus facile à utiliser et à entraîner
  - Interprétabilité : plus facile à expliquer
  - Plausibilité : rasoir d'Ockham

Problèmes à plusieurs classes



- Jeu à K classes :

$$\mathcal{X} = \{\mathbf{x}^t, \mathbf{r}^t\}_{t=1}^N$$

- Étiquettes à K dimensions :

$$\mathbf{r}^t = \left[r_1^t \; r_2^t \; \ldots \; r_K^t\right]$$

$$r_i^t = \begin{cases} 1 & \text{si } \mathbf{x}^t \in C_i \\ 0 & \text{si } \mathbf{x}^t \in C_j, j \neq i \end{cases}$$

- K hypothèses à entraîner :

$$\mathrm{h}_i,\ i = 1, \ldots, K$$

$$\mathrm{h}_i(\mathbf{x}^t) = \begin{cases} 1 & \text{si } \mathbf{x}^t \in C_i \\ 0 & \text{si } \mathbf{x}^t \in C_j, j \neq i \end{cases}$$

Régression

- Jeu de données :

$$\mathcal{X} = \{\mathbf{x}^t, r^t\}_{t=1}^N,\ r^t \in \mathbb{R}$$

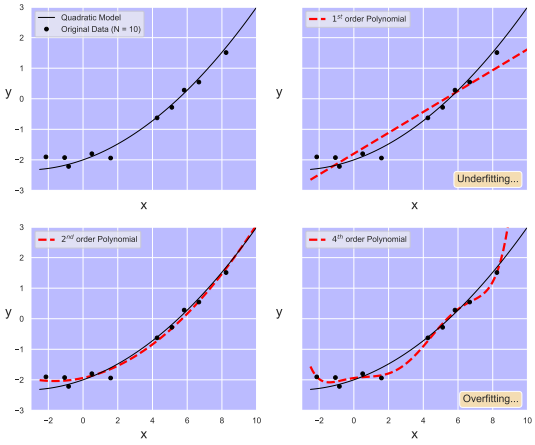
- On cherche une fonction $\mathrm{h}(\cdot)$:

$$r^t = \mathrm{h}(\mathbf{x}^t) + \epsilon$$

- Et on veut minimiser l'erreur quadratique :

$$E(\mathrm{h}|\mathcal{X}) = \frac{1}{N}\sum_{t=1}^{N}\left(r^t - \mathrm{h}(\mathbf{x}^t)\right)^2$$

Régression



- 1er ordre avec une variable :

$$\mathrm{h}(x) = w_1 x + w_0$$

- Solution avec dérivées partielles sur erreur empirique
- Sur figure, solutions avec polynômes du 1er, 2e et 4e ordre
  - 4e ordre est « presque parfait », mais généralise mal
  - 2e ordre capture mieux les données que le 1er

Sélection de modèles

- L'apprentissage supervisé est un problème *mal posé*
  - Les exemples ne sont pas suffisants pour donner une solution unique
- Il faut donc avoir un *biais inductif*, en faisant des suppositions sur $\mathcal{H}$
- Objectif premier : **généralisation**
  - Avoir le modèle qui performe le mieux sur de nouvelles données
- Sur-apprentissage : $\mathcal{H}$ est plus complexe que le concept à modéliser
- Sous-apprentissage : $\mathcal{H}$ est moins complexe que le concept

Facteurs influençant l'apprentissage

- Rappel : notre objectif est de minimiser l'erreur de généralisation sur de nouveaux exemples
- 1er facteur : complexité de la classe des hypothèses
  - Si la complexité des hypothèses augmente, alors l'erreur de généralisation diminue pendant un temps, mais ensuite augmente
- 2e facteur : taille du jeu d'exemples d'entraînement
  - Plus on a de données, plus l'erreur de généralisation diminue

Régularisation

- Régularisation : introduire une pénalité dans la fonction optimisée afin de minimiser la complexité
  - Rasoir d'Ockham : toutes autres choses étant égales, les solutions les plus simples sont les plus vraisemblables
- Forme courante : $J(\mathrm{h}) = E(\mathrm{h}|\mathcal{X}) + \lambda\, C(\mathrm{h})$
  - λ : pondération relative entre l'erreur empirique $E(\mathrm{h}|\mathcal{X})$ et la complexité $C(\mathrm{h})$ de la fonction
- Exemples de mesures de complexité utilisées pour régulariser
  - Nombre de paramètres utilisés (ou de valeurs non nulles de paramètres)
  - Norme L_2 des valeurs de paramètres
  - Dimension Vapnik-Chervonenkis
  - Degré du polynôme pour régression polynomiale

Validation empirique

- Pour estimer l'erreur de généralisation, on doit utiliser des données non vues durant l'entraînement
- Approche classique, partitionner le jeu d'exemples
  - Entraînement (50%) / validation (25%) / test (25%)
- Procédure suivie
  1. Génère des fonctions à partir du jeu d'entraînement
  2. Évalue l'erreur en généralisation de ces fonctions sur le jeu de validation, en retournant celle qui la minimise
  3. Rapporte la performance finale de la fonction choisie sur le jeu de test comme base de comparaison
- Si on a peu de données, d'autres solutions existent
  - Partitionner le jeu initial en M plis distincts
  - Utiliser $M - 1$ plis pour entraînement et le pli restant pour la validation
  - Répéter M fois, avec toutes les combinaisons possibles
  - Cas extrême : M est égal à N

Trois dimensions de l'apprentissage supervisé

- Représentation
  - Hypothèses paramétrées : $\mathrm{h}(\mathbf{x}|\theta)$
  - Instances, hyperplans, arbres de décision, ensembles de règles, réseaux de neurones, modèles graphiques, etc.
- Évaluation
  - Erreur empirique : $E(\theta|\mathcal{X}) = \frac{1}{N}\sum_{t=1}^{N}\mathcal{L}(r^t,\mathrm{h}(\mathbf{x}^t|\theta))$
  - Taux de reconnaissance, précision, rappel, erreur quadratique, vraisemblance, probabilité a posteriori, gain en information, marge, coût, etc.
- Optimisation
  - Procédure : $\theta^* = \mathrm{argmin}_{\forall\theta} E(\theta|\mathcal{X})$
  - Descente du gradient, programmation quadratique, heuristique, etc.